1. Nhap mon HTML
    1. Network In UEH
    2. Polling Network Analysis
    3. Neural Networks

Network In UEH

HƯỚNG DẪN NHẬP MÔN HTML (ĐÂY CHỈ LÀ VÍ DỤ TẠO BÀI GIẢNG)

Bài này sẽ trình bày về những bí ẩn của ngôn ngữ đánh dấu siêu văn bản (HTML), nhưng thực chất chẳng có gì bí ẩn. Nếu đã biết cách dùng trình xử lý văn bản, bạn có thể tạo ra các trang Web chỉ trong vài phút.Nhờ được xây dựng trên nền văn bản và đồ hoạ, World Wide Web (WWW) đã đưa Internet nhập vào dòng chảy của cuộc sống. Nhưng tính hấp dẫn của WWW vượt xa hơn nhiều so với giao diện đồ hoạ, là cái đã lôi cuốn nhiều người dùng PC không rành kỹ thuật. Nhờ HTML tương đối đơn giản nên nhiều người bình thường có thể tạo ra các Web site đầy ấn tượng. Nếu bạn khao khát muốn đưa cái gì đó của mình lên Web, thì tất cả những điều mà bạn cần có là một tài khoản đăng ký với hãng cung cấp Web, một ít bí quyết về HTML, và một chút óc sáng tạo.Bài báo này sẽ đưa bạn đi một chuyến du lịch trên trang Web điển hình, để bạn có thể thấy nó được thực hiện như thế nào. Đặc biệt, bạn sẽ được giới thiệu về các phần tử cấu tạo nên HTML (Hypertext Markup Language - Ngôn ngữ đánh dấu siêu văn bản) - linh hồn các trang Web. Và ngay cả trong trường hợp không có dự định tạo ra Web site riêng cho mình, thì bạn cũng có thêm được hiểu biết về những gì xảy ra khi bạn nối vào Web và xem các trang ưa thích.

HTML: Ngôn ngữ đánh dấu:

Sau những lóng ngóng với vài trang Web trong cuộc du lịch Internet, có thể bạn nghĩ rằng phải có gói phần mềm tinh vi để thu được tất cả những hiệu ứng trang trí đầy quyến rũ này. Mặc dù có một số công cụ dành cho những người say mê HTML thực sự, nhưng bạn có thể dùng một trình soạn thảo văn bản bất kỳ như Notepad của Windows để tạo ra những trang Web hấp dẫn.Sự thật là hầu hết các trang đều không có gì khác ngoài văn bản được gia cố thêm bằng một phần tử HTML xếp đặt đúng quy cách. Để hiểu được cách hoạt động của chúng, bạn phải xem xét từng từ trong cụm từ Ngôn ngữ đánh dấu siêu văn bản và xem chúng có ý nghĩa gì:Siêu văn bản (HyperText). Như bạn đã biết, liên kết siêu văn

bản là một từ hay một câu trong trang Web dùng để chỉ đến trang Web khác. Khi nhấn chuột lên một trong các liên kết này, trình duyệt của bạn (như Netscape Navigator hoặc Internet Explorer) sẽ đưa bạn tức khắc đến trang Web kia mà không cần hỏi han gì. Vì những liên kết siêu văn bản này thật sự là tính năng đặc trưng của World Wide Web, các trang Web thường được biết như là những tài liệu siêu văn bản. Cho nên HTML có từ siêu văn bản trong tên của nó, vì bạn dùng nó để tạo nên các tài liệu siêu văn bản này.Đánh dấu (Markup). Có từ điển định nghĩa markup là các chỉ dẫn chi tiết về kiểu dáng được ghi trên bản viết tay để xếp chữ in. Trong lĩnh vực này, chúng ta có thể viết lại định nghĩa như sau: các lệnh chi tiết về kiểu dáng được đánh vào tài liệu văn bản để đưa lên WWW. Đó là HTML theo một định nghĩa tóm tắt. Nó gồm một vài mã đơn giản để tạo ra văn bản nét đậm hoặc nghiêng và các bảng liệt kê dấu chấm đầu dòng, các hình đồ hoạ chèn thêm vào, cùng với sự xác định các mối liên kết siêu văn bản. Bạn đánh các mã này vào những chỗ thích hợp trong tài liệu văn bản gốc, trình duyệt Web sẽ thực hiện việc dịch chúng.Ngôn ngữ (Language). Đây có thể là từ dễ nhầm lẫn nhất trong cụm từ này. Nhiều người diễn giải HTML như là một ngôn ngữ lập trình. HTML không có gì liên quan đến việc lập trình máy tính cả. HTML gọi là ngôn ngữ chỉ vì nó gồm các tập hợp nhỏ các nhóm hai đến ba chữ và các từ mà bạn dùng để quy định kiểu dáng như nét đậm hoặc nghiêng.

Các thẻ HTML:

Bây giờ chúng ta cùng đi sâu vào các khái niệm đằng sau HTML, và xem xét một số ví dụ mẫu. Để bắt đầu, tất cả công việc phải làm là khởi động trình soạn thảo văn bản yêu thích của bạn và bảo đảm đã có một tài liệu rỗng được hiển thị.Trình xử lý văn bản của bạn phải có khả năng lưu tập tin ở dạng văn bản. Notepad làm việc này một cách tự động; còn các trình xử lý văn bản khác, như Word và WordPerfect, thì có tuỳ chọn văn bản trong hộp thoại Save As. Hãy tham khảo tài liệu thuyết minh của trình xử lý văn bản. Đồng thời bạn phải lưu các trang với đuôi mở rộng HTM hoặc HTML, như trong Homepage.htm chẳng hạn.Tạo lập trang Web là một vấn đề đơn giản, chỉ cần đánh văn bản của bạn vào rồi chèn thêm các ký hiệu đánh dấu, gọi là thẻ có dạng như sau:<TAG>văn bản chịu tác động</TAG>Phần thẻ là một mã (thường chỉ có một hoặc hai

chữ) xác định hiệu ứng mà bạn yêu cầu. Ví dụ, cho thẻ nét đậm là <B>. Cho nên nếu bạn muốn câu ACME Coyote Suppplies xuất hiện theo kiểu chữ đậm (bold), bạn phải đưa dòng sau đây vào tài liệu của mình: <B>ACME Coyote Supplies</B>Thẻ đầu tiên báo cho trình duyệt (browser) hiển thị tất cả phần văn bản tiếp theo bằng phông chữ đậm, liên tục cho đến thẻ <B>. Dấu gạch chéo (/) xác định đó là thẻ kết thúc, và báo cho trình duyệt ngưng hiệu ứng đó. HTML có nhiều thẻ dùng cho nhiều hiệu ứng khác, bao gồm chữ nghiêng (italic), dấu đoạn văn bản (paragraph), tiêu đề, tên trang, liệt kê, liên kết, và nhiều thứ nữa.

Cấu trúc cơ sở:

Các trang Web khác nhau, tẻ nhạt hay sinh động, vô nghĩa hoặc rất cần thiết, nhưng tất cả đều có cùng một cấu trúc cơ sở, nên đa số các trình duyệt đang chạy trên hầu hết các kiểu máy tính đều có thể hiển thị thành công hầu hết các trang Web.Các tập tin HTML luôn bắt đầu bằng thẻ <HTML>. Thẻ này không làm gì khác ngoài nhiệm vụ báo cho trình duyệt Web biết rằng nó đang đọc một tài liệu có chứa các mã HTML. Tương tự, dòng cuối trong tài liệu của bạn luôn luôn là thẻ </HTML>, tương đương như Hết.Chi tiết kế tiếp trong catalog thẻ HTML dùng để chia tài liệu thành hai phần: đầu và thân.Phần đầu giống như lời giới thiệu cho trang. Các trình duyệt Web dùng phần đầu này để thu nhặt các loại thông tin khác nhau về trang. Để xác định phần đầu, bạn đưa thêm thẻ <HEAD> và thẻ </HEAD> vào ngay sau thẻ <HTML>. Mặc dù bạn có thể đặt một số chi tiết bên trong phạm vi phần đầu này, nhưng phổ biến nhất là tên trang. Nếu có ai đó xem trang này trong browser, thì tên trang sẽ xuất hiện trong dải tên của cửa sổ browser. Để xác định tên trang, bạn đặt đoạn văn bản tên đó giữa các thẻ <TITLE> và </TITLE>. Ví dụ nếu bạn muốn tên trang của mình là My Home Sweet Home Page, bạn đưa nó vào như sau:<TITLE>My Home Sweet Home Page</TITLE>Phần thân là nơi bạn nhập vào các văn bản sẽ xuất hiện trên trang Web lẫn các thẻ khác quy định dáng vẻ của trang. Để xác định phần thân, bạn đặt các thẻ <BODY> và </BODY> sau phần đầu (dưới </HEAD>).Các thẻ sau đây xác định cấu trúc cơ bản của mọi trang Web:<HTML><HEAD><TITLE> tên trang <TITLE>Các thẻ tiêu đề khác</HEAD><BODY>Văn bản và các thẻ của trang Web</BODY></HTML>

Văn bản và các đoạn:

Như đã trình bày ở trên, bạn bổ sung văn bản của trang Web bằng cách đánh nó vào giữa các thẻ <BODY> và </BODY>. Tuy nhiên, cần nhớ rằng không thể bắt đầu một đoạn văn (chương, mục) mới mà chỉ ấn phím Enter. Bạn phải dùng thẻ để báo cho browser biết rằng bạn muốn chuyển vào một đoạn văn mới:<HTML><HEAD><TITLE>My Home Sweet Home Page</HEAD><BODY>Văn bản này sẽ xuất hiện trong phần thân của trang Web.<P>Văn bản này sẽ xuất hiện trong một đoạn văn bản mới</BODY></HTML>

Bổ sung định dạng và các tiêu đề:

HTML bao gồm nhiều thẻ làm đẹp cho văn bản trong trang. Bạn đã thấy ở trên, một từ hoặc một câu sẽ hiện thành dạng chữ đậm như thế nào khi được đặt vào giữa các thẻ <B> và </B>. Bạn cũng có thể biểu hiện văn bản theo kiểu chữ nghiêng bằng cách bao chúng lại bằng các thẻ <I> và </I>, và làm cho các từ xuất hiện ở dạng cách đơn với các thẻ <TT> và </TT>.Giống như các chương sách, nội dung của nhiều trang Web được chia thành các đoạn. Để giúp phân cách các đoạn này và làm cho dễ đọc hơn, bạn có thể sử dụng tiêu đề. Lý tưởng nhất, các tiêu đề này phải có tác dụng như là những đề mục nhỏ, thể hiện ý tưởng tóm tắt về đoạn văn đó. Để làm cho các đề mục này nổi bật, HTML có 6 loại thẻ tiêu đề để hiển thị văn bản theo phông chữ đậm với nhiều cỡ chữ khác nhau, từ ứng với phông lớn nhất đến ứng với phông nhỏ nhất.

Làm việc với các liên kết siêu văn bản

Thẻ HTML dùng để thiết lập các liên kết siêu văn bản là <A> và </A>. Thẻ <A> hơi khác so với các thẻ khác mà bạn đã gặp vì bạn không thể dùng nó một mình mà phải kèm thêm địa chỉ của tài liệu bạn muốn liên kết. Sau đây là cách hoạt động của nó:<AHREF=Address>HREF là viết tắt của hypertext reference (tham chiếu siêu văn bản). Chỉ cần thay từ địa chỉ bằng địa chỉ thực của trang Web mà bạn muốn dùng để liên kết . Dưới đây là một ví dụ: <AHREF=http://www/dosword.com/dosworld/index.html>Như vậy còn

chưa kết thúc. Tiếp theo, bạn phải cung cấp một số văn bản diễn giải liên kết để nhấn chuột vào đó. Công việc còn lại là chèn văn bản vào giữa các thẻ <A> và </A> như sau:<AHREF=address> Văn bản liên kết </A>Sau đây là một ví dụ :Why not head to the <AHREF=http://www.dosworld.com/dosworld/index.html>DOSWorld home page</A>?

Chèn hình:

Nếu bạn muốn Web site của mình nổi trội hơn, bạn phải theo xu hướng đồ hoạ với các hình ảnh được chọn kỹ lưỡng. Làm thế nào để chèn hình vào trong khi các tập tin HTML chỉ có văn bản? Nhờ bổ sung thẻ <IMG> vào tài liệu, nó sẽ ra lệnh Chèn một hình vào đây. Thẻ này xác định tên của tậ tin đồ hoạ để trình duyệt có thể mở tập tin đó và hiển thị các hình:<IMG SRC=filename>ở đây, SRC là viết tắt của source (nguồn) và tên tập tin là tên và đường dẫn của tập tin đồ hoạ mà bạn muốn hiển thị (dùng dạng GIF hoặc JPG). Giả sử bạn có một hình tên là logo.gif nằm trong thư mục Graphics. Để đưa nó vào trang Web, bạn dùng dòng sau đây.<IMG SRC=Graphics/logo.gif>

Các bảng tham khảo HTML trên Web:

Một vài thẻ bạn thấy trong bài này chỉ là những cái vụn vặt trên bề mặt HTML. Có hàng tá các thẻ bổ sung khác cho phép bạn thành lập các danh sách liệt kê dấu đầu dòng, các bảng, và cả các biểu tương tác.Để tìm hiểu về chúng, bạn hãy thử dùng một trong các bảng tham khảo HTML trên Web. Yahoo! cung cấp một danh sách các tham khảo này ở địa chỉ http://www.yahoo.com/Computers_and_Internet/ [link] Information_and_Documentation/Data_Formats/HTML/Reference/Đồng thời, Microsoft cũng có một bảng tham khảo HTML xuất sắc (dĩ nhiên đã được cài vào Internet Explorer của riêng họ) ở địa chỉ: http://www.microsoft.com/workshop/author/newhtml/default.html. [link]Cuối cùng khi bạn đã trở thành chuyên gia viết mã HTML, thì một trong các phương pháp tốt nhất để biết nhiều hơn về HTML là xem mã mà những người khác đã dùng để xây dựng các trang của họ. Ngay khi có một trang cụ thể trong browser của mình, bạn hãy lưu chúng lại (dùng

File/Save As trong Netscape Navigator hoặc File/Save As File trong Internet Explorer), rồi mở tập tin đã lưu nó ra xem trong trình xử lý văn bản của bạn.Ngoài ra, Netscape Navigator và Internet Explorer cũng cho phép xem các trang bất kỳ được mở ra trong browser; hãy chọn View/Document Source trong Netscape Navigator hoặc View/Source trong Internet Explorer.Các bộ soạn thảo HTMLKhông nhất thiết phải tìm hiểu các thẻ HTML để tạo dựng các trang Web. Có nhiều chương trình Windows có thể thực hiện công việc chèn thêm các thẻ thích hợp này một cách tự động. Sau đây là ví dụ một số trình soạn thảo đang có sẵn:Word 97. Phiên bản mới nhất của Microsoft Word có sẵn khả năng HTML, bao gồm cả các ví dụ mẫu và lệnh Save to HTML để chuyển đổi các tài liệu World hiện hữu thành HTML.Netscape Composer. Một phần của bộ Netscape Communicator, có kỹ thuật tạo trang WYSIWYG (thấy gì được nấy) và tích hợp chặt chẽ với Netscape Navigator. Hãy tìm địa chỉ http://home.netscape.com/để có nhiều thông tin hơn.WebEdit. Chương trình WebEdit xuất sắc của Kenn Nesbitt có giao diện trực giác và hỗ trợ hầu hết các thẻ HTML có trên hành tinh này, đồng thời tốc độ rất nhanh. HomeSite. Đây là một trình biên tập HTML tương đối mới. Một điều rất ngạc nhiên là nó được biên soạn bởi một người vẽ tranh biếm hoạ - tác giả của phim hoạt hình Dexter. Nó là một bộ soạn thảo đầy đủ tính năng, bao gồm cả trình kiểm tra chính tả cài sẵn, browser, frame wizard, các thẻ HTML mã hoá màu, và nhiều cái khác nữa. Tìm theo địa chỉ http://www.dexnet.com/homesite.html. [link]Bùi Xuân ToạiDOS World 6/97

Polling Network Analysis

A polling network is a computer communications network that uses **polling** to control access to the network. Each node or station on the network is given exclusive access to the network in a predetermined order. Permission to transmit on the network is passed from station to station using a special message called a **poll**. Polling may be centralized (often called **hub polling**) or decentralized (**distributed**). In hub polling, the polling order is maintained by a single central station or **hub**. When a station finishes its turn transmitting, it sends a message to the hub, which then forwards the poll to the next station in the polling sequence. In a decentralized polling scheme, each station knows its successor in the polling sequence and send the poll directly to that station. To simplify matters, we will assume a distributed polling scheme.

The analysis of a polling network uses the results of the analysis of an M/G/1 queue with vacations. Each vacation corresponds to the transfer of the poll from one station to the next in the polling cycle. We divide time into alternating types of intervals: **polling intervals**, during which the poll is transferred between stations, and **transmission intervals**, during which the station with the poll transmits packets.

Polling networks come in three flavors: gated, exhaustive, and partially gated. In a **gated** system, each station is allowed to transmit only those packets that arrived prior to the start of the poll interval (i.e., prior to the start of the vacation preceding the station's use of the network). An **exhaustive** scheme allows a station to transmit any packets that arrive before it transfers the poll to the next station. A **partially gated network** allows stations to transfer all packets that arrive by the time the poll does. Polling networks will typically be partially gated or exhaustive, not gated.

We assume that arrivals at each of the m stations are independent Poisson processes with rate $\frac{\lambda}{m}$. Note that "arrival" refers to a message arriving from the "outside world" to a station in order to be transmitted over the network; it does not mean the arrival of a message that has been transmitted over the network.

Example:
gated system, m=1
We define the following notation:

- $\overline{X}$ - mean packet length in seconds
- $\overline{X^2}$ - second principal moment of the packet length distribution
- V_r - r.v. for length of r^{th} polling interval. $\{V_r\}$ are independent and identically distributed.
- $\overline{V} = E[V_r]$
- $\overline{V^2} = E\left[{V_r}^2\right]$
- R_i - r.v. for residual time data packet i must wait in queue until end of current packet transmission or polling interval
- $V_{r(i)}$ - r.v. for length of polling interval for data packet i
- $\rho = \lambda\overline{X}$

A packet which arrives in a gated system with one station must first wait the residual service time for the packet currently being transmitted or the residual length of the poll transfer time, depending on when it arrives. It must then wait until all of the packets queued for transmission at its arrival have been serviced. Finally, it must wait until the next poll transfer is finished (note that in this simplistic system, the station sends the poll to itself).

$$E[W_i] = E[R_i] + E[N_i]\overline{X} + E\left[V_{r(i)}\right]$$

The time-average for the residual service time can be obtained as it was for the M/G/1 queue with vacations.

$$E[R_i] = \frac{\lambda\overline{X^2}}{2} + \frac{(1-\rho)\overline{V^2}}{2\overline{V}}$$

Also as before,

$$\underset{i\to\infty}{\text{limit}}\ E[N_i]\overline{X} = \underset{i\to\infty}{\text{limit}}\ \lambda E[W_i]\overline{X} = \rho W$$

$$\underset{i\to\infty}{\text{limit}}\ E\left[V_{r(i)}\right] = \overline{V}$$

$$W = \frac{\lambda\overline{X^2}}{2} + \frac{(1-\rho)\overline{V^2}}{2\overline{V}} + \rho W + \overline{V} = \frac{\lambda\overline{X^2}}{2\times(1-\rho)} + \frac{\overline{V^2}}{2\overline{V}} + \frac{\overline{V}}{1-\rho}$$

If $V_i = A$, a constant for all i, this can be simplified to

$$W = \frac{\lambda\overline{X^2}}{2\times(1-\rho)} + \frac{A^2}{2A} + \frac{A}{1-\rho} = \frac{\lambda\overline{X^2}}{2\times(1-\rho)} + \frac{A\,(1-\rho)+2A}{2\times(1-\rho)} = \frac{\lambda\overline{X^2}}{2\times(1-\rho)} + \frac{A}{2}\frac{3-\rho}{1-\rho}$$

Example:
m>1
The case of one station has relatively little practical application in computer networks, but the analysis does serve as a convenient starting point for $m > 1$ stations. We first need to define some additional notation:

- Y_i - r.v. for combined length of all of the whole polling intervals during which packet i must wait
- N_i - r.v. for total number of packets that must be transmitted after the arrival of packet i and before i is transmitted (not including any packet in service when i arrives)
- R_i - r.v. for residual time for the packet or poll in progress
- $Y = \underset{i\to\infty}{\text{limit}}\ E[Y_i]$

A packet must wait

1. while the packet transmission or poll interval underway at its arrival finishes;
2. for all packets which arrived before it but which had not been serviced yet to be transmitted;
3. for the time required to transfer the polls from station to station until the transmission interval in which the packet will be transferred starts.

$$E[W_i] = E[R_i] + E[N_i]\overline{X} + E[Y_i]$$

As before

$$R = E[R_i] = \frac{\lambda \overline{X^2}}{2} + \frac{(1-\rho)\sum_{r=0}^{m-1} \overline{V_r}^2}{2\sum_{r=0}^{m-1} \overline{V_r}}$$

Each packet transmitted before i has an average transmission time $\overline{X}$. N_i is not the number of packets ahead of i in the queue at its arrival, since packets might arrive at other stations after i arrives, but actually be transmitted before i because of the polling cycle. However, by Little's Theorem,

$$\operatorname*{limit}_{i \to \infty} \; E[N_i]\overline{X} = \lambda W \overline{X} = \rho W$$

Consequently,

$$W = R + \rho W + Y = \frac{R + Y}{1 - \rho}$$

Y depends on the flavor (gated, partially gated, or exhaustive) of polling network. We will look at each.

Exhaustive System

α_{jk} is the expected value of Y_i given that packet i arrives in user j's polling or data interval and belongs to user $(j+k) \bmod m$.

$$\alpha_{jk} = \begin{cases} 0 \;\text{ if }\; k = 0 \\ \overline{V_{(j+1)\mathrm{mod}m}} + \ldots + \overline{V_{(j+k)\mathrm{mod}m}} \;\text{ if }\; k > 0 \end{cases}$$

We first remove the condition that the packet belongs to station $(j+k) \bmod m$ by assuming that a packet belongs to a particular station with probability $\frac{1}{m}$ for all stations. The expected value of Y_i given that packet i arrives in user j's polling or data interval is given by

$$\frac{1}{m}\sum_{k=1}^{m-1} \alpha_{jk} = \sum_{k=1}^{m-1} \frac{m-k}{m} \overline{V_{(j+k)\mathrm{mod}m}}$$

Since all users are identical, they have equal average length data intervals in steady-state, and the steady-state probability that a packet arrives in a particular user's data interval is $\frac{\rho}{m}$. Similarly, the probability that a packet arrives during a particular user's polling interval is $\frac{(1-\rho)\overline{V_r}}{\sum_{k=0}^{m-1}\overline{V_k}}$.

Equation:

$$\begin{aligned} Y &= \sum_{r=0}^{m-1}\left(\frac{\rho}{m}+\frac{(1-\rho)\overline{V_r}}{\sum_{k=0}^{m-1}\overline{V_k}}\right)\sum_{j=1}^{m-1}\frac{m-r}{m}\overline{V_{(r+j)\text{mod}m}} \\ &= \frac{\rho}{m}\sum_{r=0}^{m-1}\sum_{j=1}^{m-1}\frac{m-j}{m}\overline{V_{(r+j)\text{mod}m}}+\frac{1-\rho}{\sum_{k=0}^{m-1}\overline{V_k}}\sum_{r=0}^{m-1}\overline{V_r}\sum_{j=1}^{m-1}\frac{m-j}{m}\overline{V_{(r+j)\text{mod}m}} \\ &= \frac{\rho}{m}\sum_{r=0}^{m-1}\frac{m-j}{m}\sum_{j=1}^{m-1}\overline{V_{(r+j)\text{mod}m}}+\frac{1-\rho}{\sum_{k=0}^{m-1}\overline{V_k}}\sum_{r=0}^{m-1}\sum_{j=1}^{m-1}\frac{m-j}{m}\overline{V_r}\overline{V_{(r+j)\text{mod}m}} \end{aligned}$$

Equation:

$$\begin{aligned} \sum_{r=0}^{m-1}\sum_{j=1}^{m-1}\frac{m-j}{m}\overline{V_r V_{(r+j)\text{mod}m}} &= \sum_{r=0}^{m-1}\sum_{j=0}^{m-1}\frac{m-j}{m}\overline{V_r V_{(r+j)\text{mod}m}}-\sum_{r=0}^{m-1}\overline{V_r^{\,2}} \\ &= \frac{1}{2}\left(\left(\sum_{r=0}^{m-1}\overline{V_r}\right)^2-\sum_{r=0}^{m-1}\overline{V_r^{\,2}}\right) \end{aligned}$$

The mean polling interval length averaged over all users is

$$\overline{V}=\frac{1}{m}\sum_{r=0}^{m-1}\overline{V_r}$$

Thus

Equation:

$$\begin{aligned} Y &= \frac{\rho}{m}\sum_{j=1}^{m-1}(m-j)\overline{V}+\frac{1-\rho}{2m\overline{V}}\left(\left(m\overline{V}\right)^2-\sum_{r=0}^{m-1}\overline{V_r^{\,2}}\right) \\ &= \frac{\rho\overline{V}}{m}\sum_{j=1}^{m-1}m-j+\frac{(1-\rho)m\overline{V}}{2}-\frac{(1-\rho)\sum_{r=0}^{m-1}\overline{V_r^{\,2}}}{2m\overline{V}} \\ &= \frac{\rho\overline{V}(m-1)}{m}+\frac{(1-\rho)m\overline{V}}{2}-\frac{(1-\rho)\sum_{r=0}^{m-1}\overline{V_r^{\,2}}}{2m\overline{V}} \\ &= \frac{(m-\rho)\overline{V}}{2}-\frac{(1-\rho)\sum_{r=0}^{m-1}\overline{V_r^{\,2}}}{2m\overline{V}} \end{aligned}$$

Substituting these expressions for R and $\overline{V}$ into the equation for W,

Equation:

$$\begin{aligned} W &= \frac{\lambda\overline{X^2}}{2\times(1-\rho)}+\frac{\sum_{r=0}^{m-1}\overline{V_r^{\,2}}}{2m\overline{V}}+\frac{(m-\rho)\overline{V}}{2\times(1-\rho)}-\frac{\sum_{r=0}^{m-1}\overline{V_r}^{\,2}}{2m\overline{V}} \\ &= \frac{\lambda\overline{X^2}}{2\times(1-\rho)}+\frac{(m-\rho)\overline{V}}{2\times(1-\rho)}+\frac{\sum_{r=0}^{m-1}\overline{V_r^{\,2}}-\overline{V_r}^{\,2}}{2m\overline{V}} \end{aligned}$$

Thus, for exhaustive gating,

$$W = \frac{\lambda \overline{X^2}}{2 \times (1-\rho)} + \frac{(m+\rho)\overline{V}}{2 \times (1-\rho)} + \frac{{\sigma_V}^2}{2\overline{V}}$$

Partially gated system

In the partially gated system, a packet that arrives during a user's own data interval is delayed by an additional $m\overline{V}$ on average, and this occurs with probability $\frac{\rho}{m}$, thus increasing Y by $\rho\overline{V}$ compared to the exhaustive case.

$$W = \frac{\lambda \overline{X^2}}{2 \times (1-\rho)} + \frac{(m+\rho)\overline{V}}{2 \times (1-\rho)} + \frac{{\sigma_V}^2}{2\overline{V}}$$

Fully gated system

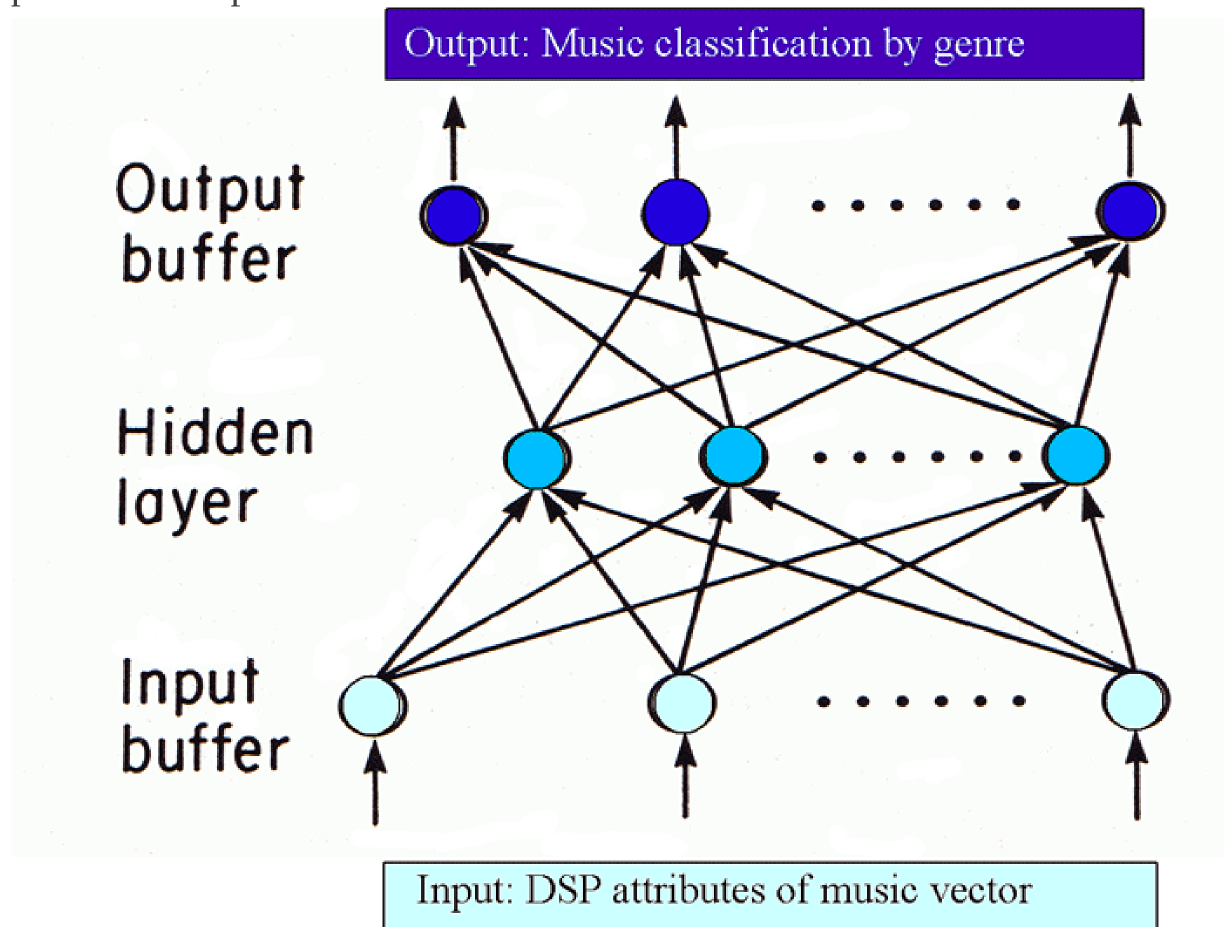
If the system is fully gated, then a packet that arrives during a user's own polling interval is also delayed by an average of $m\overline{V}$, and this is in addition to the extra delay incurred in the partially gated case. The probability of this occurring is $\frac{1-\rho}{m}$, thus increasing Y by $(1-\rho)\overline{V}$ compared to the partially gated case.

$$W = \frac{\lambda \overline{X^2}}{2 \times (1-\rho)} + \frac{(m+2-\rho)\overline{V}}{2 \times (1-\rho)} + \frac{{\sigma_V}^2}{2\overline{V}}$$

Neural Networks

At their core, neural networks are pattern recognition systems. They predict an output given a sequence of inputs and their corresponding classification. They are based on biological nervous systems, in which there are many inputs and numerous outputs to a single neuron. On the highest level, the neural network is a primitive learning machine that can be used to process data such as stock market quotes, DNA sequences, and in our case, music classification.

Neural networks are systems that take a lengthy input, process the data, and predict an output.



The processing is done by multiple, weighted layers of nodes. Each node is connected to every node in the next layer, and at each interface between nodes are connecting fibers weighted by a sum. Neural networks are given a vector of inputs, usually longer than the output. The first layer of nodes is

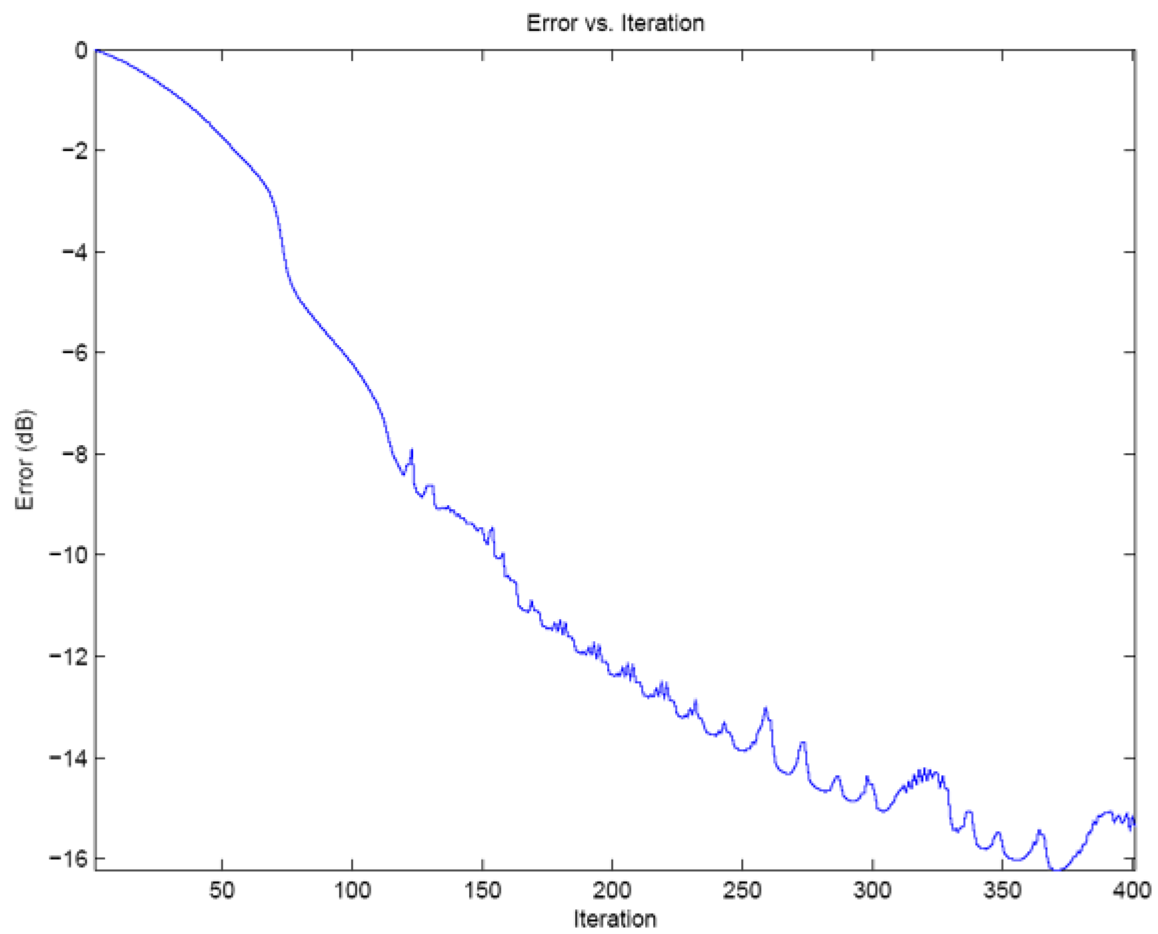
the same length as the input. The nodes at each successive layer sum their inputs, weight the sum, and produce an output. The output of the final layer is the output of the system. In this manner, an output is predicted given an input.

The remaining question is how the weights are determined. The use of neural networks is twofold: you must first "train" the network by giving it inputs and their corresponding outputs, and then you may test the network by giving it inputs with no outputs. The training determines the weighting on the nodes. For example, we train the neural network by giving it the vectors of signal processing data (bandwidth, power spectral density, etc.) as well as the corresponding classification of music. Classical music is denoted as [1 0 0 0 0 0], jazz is denoted as [0 1 0 0 0 0], etc., as shifted delta functions.

There are many methods to train neural networks, but the one we use is called backpropogation. The neural network takes the input and feeds it through the system, evaluating the output. It then changes the weights in order to get a more accurate output. It continues to run the inputs through the network multiple times until the error between its output and the output you gave it is below a defined tolerance level.

After training is completed, we test the neural network with songs that it has never seen before. It predicts a classification of genre based on the weights it created during training.

To train the neural network, we used the method of back propagation. There were 15 nodes in the hidden layer, and we used an adaptive learning rate training function. This means that the network analyzed its learning rate after each iteration, changing it to remain relatively constant. For instance, if the learning curve is too steep, and the network is learning too quickly, it decreases its learning rate, and vice-versa. This is a graph of the error (learning rate) versus time:



Error in the neural network decreases with each successive iteration.